The Virtual Sailor: An Implementation of Interactive Human Body Modeling *

David T. Chen[1] Steven D. Pieper[1,2] Sunil K. Singh[2]
Joseph M. Rosen [1,3] David L. Zeltzer[4]
[1] Medical Media Systems, [2] Thayer School of Engineering, Dartmouth College,
[3] Dartmouth Hitchcock Medical Center,
[4] Media Laboratory and Research Laboratory for Electronics, MIT.

1 Introduction

The primary focus of the Virtual Sailor project is the creation of 'virtual actors', for training and other applications where human figures must be coordinated. By the term virtual actor, we refer to computer animated figures that can mimic human beings in physical form, function and behavior. In this paper, we discuss the underlying techniques and software systems for creating the "skin, bone, and muscles" of aspects of interactive human body models.

Our approach combines a kinematic description of the human skeletal system with finite element model of the skin and soft tissue of the face. We present a set of numerical techniques that allow us to implement these models efficiently so that they can be used in a real-time simulation environment. To animate these figures, we are formulating a set of control systems to mimic basic human behaviors.

Such software systems can be used as 'inhabitants' of virtual environments that simulate real world scenarios. Our complete system will combine the body models described here with a task-level motor programming system described elsewere [21]. In another domain, video games provide simplified agents that can carry out certain stylized behaviors in response to user actions. However, these figures are presented as 2D bitmaps, rather than accurate 3D models. We can easily imagine creating a three dimensional simulation of a basketball court using virtual environment technology.

Realistic training environments may consist of intricate models of machinery and several humans. Creating these complex worlds requires software tools which allow the incremental definition of pieces which can then be assembled into a complete system. We use a world description language which allows us to modularlize our software design and add new capabilities as they are developed. The virtual environment programming environment which we are using for this research is called 3d. This is a VE "testbed" system to support the specification of behaviors for virtual worlds, the development of interactive simulations, and the design and implementation of VE interfaces. (see [3] for details).The heart of the 3d system is the dialog manager [1]. Commands are accepted from the key- board, through X events, or from manipulator programs [12]- including I/O device controllers and VE simulation modules-that communicate via UNIX pipes. These com- mands create and control an extensible, typed object database that defines the state of the synthetic world.

The major topics that we present in this paper are the following:

- *Kinematic models of rigid bodies.* In section 2, we outline the general framework for describing coupled kinematic chains. We use this framework to model various articulated joints in the human body. In particular, we describe how this approach is used to model the upper limbs and head in section 2.

*This research is supported by the Office of Naval Research.

- *Control systems.* The coordination of multiple joints or sub-systems must be accomplished by various control systems. Broadly speaking, motion controllers may be described as *kinematics based* (encompassing geometric information only) or *dynamics based* (which include the dynamics of the systems being controlled or coordinated). We describe a *kinematics based* rigid body controller in Section 3 for the motion control of the upper limbs and the head. An example of a controller based on dynamic models of the system is described in Section 4 to coordinate eye motions.

- *Soft tissue deformations.* The outer skin and the underlying tissue must be treated as a continuum necessitating models based on continuum mechanics. Facial expressions are an important communication channel In section 5 we describe our implementation of a facial tissue simulator that combines a sophisticated physical model with real-time performance.

2 Kinematics of Human Body Joints Modeled as Rigid Bodies

A static image may be accurately captured by a reasonably fine polyhedral approximation. To display rigid bodies in kinematic motion, we must model the geometry and the geometric constraints between two links which are connected. To display realistic dynamic simulations, we must also account for mass and inertia properties of body segments. Several researchers have experimented with efficient dynamic simulation algorithms [9, 7, 19].

Using the formalism developed by mechanical engineers and robotics researchers, jointed figures generally are considered to be networks of linked manipulators. The legs and arms of a human skeleton, for instance, can be described as manipulators attached to a common fixed reference frame centered on the body. The hands and feet in this case play the role of *end effectors*.

A very common coordinate frame representation in use is that of Denavit and Hartenberg [10]. A single DH quadruple is used to represent each degree of freedom of the manipulator. Most joints in a human body may be described using this formalism. We briefly describe the human upper arm and head below.

The human upper arm

The human upper arm can be idealized as a kinematic system composed of five links and eleven degrees of freedom, and so requires eleven DH quadruples to represent the joint coordinate frames. These eleven degrees of freedom correspond to three DOFs each at the clavicle, scapula and humerous and one DOF at both the ulna and radius.

The head

The head rests on the neck. We model the neck as a series of joints, each joint capable of limited motion about three axes. At the end of the chain is the head and its kinematics can be fairly accurately represented by treating it like a *spherical wrist* in robot manipulators. The limited motion of the neck joints allows *restricted* positioning of the head in space. In addition, there are three degrees of freedom (the roll, pitch and yaw) for the head itself and the three coordinate axes intersect at a point. The Euler angles may be used to represent these motions of the head. Thus if the motion of the head is represented by the triple (α, β, γ) which represents, respectively, rotations about the *current* Z, Y, Z axes, the resultant orientation of the head can be represented by the matrix

$$R = \begin{bmatrix} c\alpha c\beta c\gamma - s\alpha s\gamma & -c\alpha c\beta s\gamma - s\alpha c\gamma & c\alpha s\beta \\ s\alpha c\beta c\gamma + c\alpha s\gamma & -s\alpha c\beta s\gamma + c\alpha s\gamma & s\alpha s\beta \\ -s\beta c\gamma & s\beta s\gamma & c\beta \end{bmatrix}$$

3 Motion controllers based on kinematics only

Kinematics-based motions satisfy all geometric constraints that are imposed on them but contain no information about the dynamics of the systems involved.

3.0.1 Inverse kinematic control of the upper arm of a graphical robot

We have implemented a graphical robot whose motions are controlled by inverse kinematic techniques, which compute the motions of its shoulder, elbow, and wrist joints so that the hand moves to specific targets in space-such as the representation of the user's hand. The user guides the high level activity of robot and directly controls the motion with her or his own hand. The kinematic structure of the arm is described through Denavit-Hartenberg joint notation [10].

In a similar manner, the *Head Control Unit* uses a purely inverse kinematics-based controller. The desired orientation of the head is computed by determining the normal to the object. This specifies the *final* orientation matrix R above. Given R as

$$R = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix}$$

The corresponding $Z-Y-Z$ Euler angles that would lead to this orientation are then determined as follows. If $\sin(\beta) \neq 0$ (i.e. *both* r_{31} and r_{32} are not zero), then

$$\begin{aligned} \beta &= \text{Atan2}\left(\sqrt{r_{31}^2 + r_{32}^2}, r_{33}\right) \qquad (1) \\ \alpha &= \text{Atan2}\left(\frac{r_{23}}{s\beta}, \frac{r_{13}}{s\beta}\right) \\ \gamma &= \text{Atan2}\left(\frac{r_{32}}{s\beta}, \frac{-r_{31}}{s\beta}\right) \end{aligned}$$

Since we always pick the positive square root, we only pick solutions that satisfy $0 \leq \beta \leq 180$. If $\sin(\beta) = 0$ then $\beta = 0.0$ and the above mentioned solutions degenerate. In this case we simply pick $\alpha = \beta = 0.0$ and

$$\gamma = \text{Atan2}(-r_{12}, r_{11})$$

The head is now rotated by angles α, β, γ about the $Z-Y-Z$ axes. An important restriction that must be imposed on this controller is the *joint limits* on all three rotations. This is essential to prevent mathematically correct but anatomically infeasible head motions (such as turning the head around by 180 degrees).

4 Motion Control Based on Dynamic Models

An instance where we have explicitly accounted for the dynamics of the object and the human system is in our implementation of the *eye control unit.* This control unit is responsible for the motions of the eye-balls of the virtual actor in response to a target. The implementation of this controller has been based on experimental data from measured responses of human subjects [6]. In particular, the eye control unit accounts for the following observed features in the human system:

- **Voluntary saccades.** These refer to quick redirections of the line of sight towards the object of interest. There is usually a delay of about 200 msec from the stimulus for a saccade until its enactment.

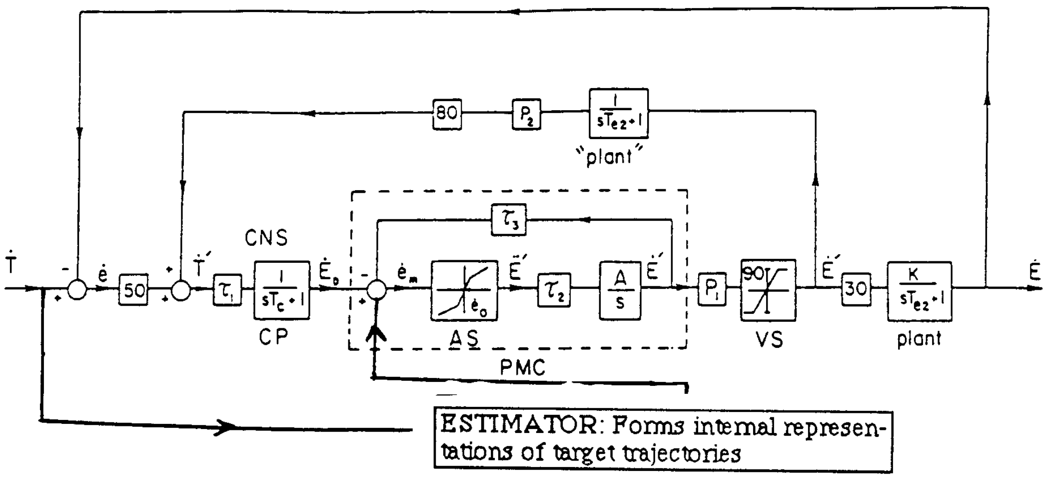

Figure 1: A block diagram of our modified control system showing predictive compensation for smooth pursuit.

- **Smooth pursuit.** Human beings display the ability to track with *zero latency* (no error) targets that are in predictable motion. Standard feedback controllers which are driven purely by position and velocity errors cannot account for this. . To capture this ability we have developed a *predictive controller* which can estimate trajectory parameters and use these to drive the eye movements. In addition, we have included a nonlinear velocity and acceleration saturation feature in the controller

Figure 1 shows the block diagram of the widely accepted model of the control system for the eye [16] which we have implemented. Some special features of this model are:

- The internal delays, shown as τs in various boxes.
- The use of the target velocity $\dot{T}$ as the driving reference signal.
- A velocity saturating unit (VS) and an acceleration saturation unit (AS) to account for various nonlinearities such as overshoot and ringing and to account for the degradation in tracking faster trajectories.
- Use of an *internal positive feedback.* This is a copy of the motor command known as the *efference copy* and was introduced to account for several inconsistencies in earlier models.

It has been shown [6] that for predictable motions the eyes track the target with zero latency. To account for this, we have introduced a *predictor* as shown in Figure 1. This predictor adaptively estimates the trajectory based on an input-output model (similar to a Kalman filter) and provides corrective compensation to the motor command resulting in zero latency. We also plan to implement *combined head-eye movements* and a controller for binocular coordination of eye motions (vergence eye movements) in the future.

5 Facial Animation: An Example of Simulation Based on Elastic Body Dynamics

Early work on facial animation was based on parameterized surface representations of the human face [11]. Several computer animated sequences have portrayed characters that exhibit effective,

parameter	type	meaning
p_o	real 3 vector	origin point
r_o	real number	origin radius
p_i	real 3 vector	insertion point
r_i	real number	insertion radius
l	real number	muscle load

Table 1: Parameters of the facial muscle model.

albeit somewhat stylized, human facial expressions [2, 4]. Methods have been developed to control facial expressions using muscle models [15, 18] or abstract representations of muscle [8, 17]. These systems operate on surface representations, and most make use of the Facial Action Coding System (FACS) [5] which provides a catalog of human facial expressions, and a mapping from muscle action to resulting facial expressions. Good results have been obtained, and realtime animation is possible. The most realistic simulation, however, is obtained using the discrete simulation method (DSM) or the finite element method (FEM) to simulate the multi-layered structure of human facial tissue, combined with simulated muscle action. Our technique for this project is based on Pieper's FEM based model [14] which uses isoparametric interpolation functions as deformation functions. Pieper introduced the use of FEM, since it is more amenable to volumetric analysis of various loading conditions, and it is easier to vary the sizes and shapes of the elements [12, 14]. While FEM requires lengthy set-up to compute stiffness matrices and other quantities, these are onetime computations that need not be redone unless the model is re-tesselated or the resolution of the finite elements is altered. Simulation of muscle action and the resultant deformations amount to solving a linear system, which can be computed in realtime or near realtime on current workstations.

The major advantages of this technique are the accuracy of the model, and the efficiency with which it can be calculated. The accuracy of the method is due to the fact that the use of volumetric finite elements to represent soft tissue. This provides much better results than mass-spring systems which have recently come into use in facial animation. The method is particularly efficient because the finite element stiffness matrix is constant through the course of the animation; thus it only needs to be decomposed once. This decomposed matrix can then be used repeatedly to evaluate the deformation of soft tissue under a variety of loading conditions. In particular, if we know the muscles responsible for a particular expression, then the loads are applied as discussed in the following section.

5.1 Control of Muscle Forces

A particular load condition of importance for simulation of soft tissue is loading due to muscle action in the tissue. Facial muscles are closely meshed with the other soft tissue and do not act with single points of attachment. To model this effect, we use a body force calculation that assumes a spherical volume of muscle attachment over which the force is applied. The muscle is described by five parameters given in Table 1. Muscle parameters are given in world space so that the same parameters can be used for different finite element meshes.

At a given point q, the muscle force vector f_m^i corresponding to the action of muscle i is given by:

$$f_m^i(q) = w(q, p_i^i, r_i^i)l^i(p_o^i - p)i^i) + w(q, p_o^i, r_o^i)l^i(p_i^i - p_o^i) \tag{2}$$

where w is a weighting function which scales the force at a point based on its distance from the

Figure 2: Some examples of facial expresssions from muscle actions

point of origin or insertion. For the spherical muscles used in the our current system, this function is given by:

$$w(q,p,r) = \begin{cases} 1 & \text{if } \sqrt{q \cdot p} < r \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

Other functions w could be used, for example, to make the muscle force fall off over a region based on the distance from the attachment point. For muscles which have their origin in bone, and thus generate no force on the soft tissue, the parameter r_o is set to zero. The muscle force vector for an element m is given by:

$$R_m = \sum_{\mathrm{i,j,k}=1}^{3} \alpha_{i,j,k} H^{(m)T} \sum_{a=1}^{N_m} f_m^a(x(r_i, s_j, t_k)) \tag{4}$$

where N_m is the number of muscles acting on the structure. A number of muscle load functions may be combined to approximate the effect of complete muscles. Figure 2 shows the effect of such muscle-actions on the polyhedral representation generated from Cyberware scan.

We expect to continue our efforts and integrate these features, while adding several other layers and details. These will include a control system to apply desired time-varying loads to the facial muscles to achieve a desired expression, and cordinating it with the head and eye controllers. There is still much work to be done in this field but we have obtained promising results so far.

References

[1] Appino, P. A., Lewis, J. B., Koved, L., Ling, D. T., Rabenhorst, D. A. and Codella, C. F., *An Architecture for Virtual Worlds*, Presence: Teleoperators and Virtual Environments Vol 1., No. 1., pp. 1-17, March 1992.

[2] Bergeron, P. and P. Lachapelle. *Controlling Facial Expressions and Body Movements* in the Computer Generated Animated Short Tony de Peltrie in Course Notes, Advanced Computer Animation, SIGGRAPH 85, San Francisco CA.

[3] Chen, D.T. and D. Zeltzer. *Pump It Up: Computer Animation of a Biomechanically Based Model of Muscle Using the Finite Element Method* in Proc. Siggraph 92, July 27-31, 1992, Chicago IL, pp. 89-98.

[4] deGraf, B. *Notes on Human Facial Animation* in Course Notes, State of the Art in Facial Animation, SIGGRAPH 89, July 31, 1989, Boston MA, pp. 10-11.

[5] Ekman, P. and W. Friesen, **Facial Action Coding System.** 1978, Palo Alto CA: Consulting Psychologists Press.

[6] Hughes, H. *Personal Communication*, Dartmouth College, 1992.

[7] Isaacs, P.M. and M.F. Cohen, *Controlling Dynamic Simulation with Kinematic Constraints*, Behavior Functions and Inverse Dynamics. Proc. Siggraph 87, July 1987, 21(4), pp. 215-224.

[8] Magnenat-Thalmann, N., N. Primeau, and D. Thalmann, *Abstract Muscle Action Procedures for Human Face Animation.* The Visual Computer, January 1988, 4(1).

[9] McKenna, M., S. Pieper, and D. Zeltzer. *Control of a Virtual Actor: The Roach* in Proc. 1990 Symposium on Interactive 3D Graphics, March 25-28, 1990, pp. 165-174.

[10] Paul, R. . **Robot Manipulators: Mathematics, Programming, and Control**. MIT Press, 1981.

[11] Parke, F.I., *A Parametric Model for Human Faces*, Tech. Report UTEC-CSc-75- 047, 1974, University of Utah: Salt Lake City.

[12] Pieper, S. *CAPS: Computer-Aided Plastic Surgery, Ph.D. Thesis*, September 1991, Massachusetts Institute of Technology.

[13] Pieper, S., *More Than Skin Deep: Physical Modeling of Facial Tissue M.S. Thesis*, February 1989, Massachusetts Institute of Technology.

[14] Pieper, S., J. Rosen, and D. Zeltzer. *Interactive Graphics for Plastic Surgery: A Task-Level Analysis and Implementation* in Proc. 1992 Symposium on Interactive 3D Graphics, March 29-April 1, 1992, Cambridge MA: ACM Press, pp. 127-134.

[15] Platt, S.M. and N.I. Badler, *Animating Facial Expressions.* Computer Graphics, August 1981, 15(3), pp. 245-252.

[16] Robinson, D.A., Gordon, J. and Gordon, S. *A Model of the Smooth Pursuit Eye Movement System.* Biological Cybernetics, Vol. 55, pp. 43-57, 1986.

[17] Waite, C.,*The Facial Action Control Editor, FACE: A Parametric Facial Expression Editor for Computer Generated Animation, M.S.V.S Thesis*, December 1988, Massachusetts Institute of Technology.

[18] Waters, K., *A Muscle Model for Animating Three-Dimensional Facial Expression.* Computer Graphics, July 1987, 21(4).

[19] Wilhelms, J., *Using Dynamic Analysis for Realistic Animation of Articulated Bodies* . IEEE Computer Graphics and Applications, June 1987, 7(6), pp. 12-27.

[20] Zeltzer, D., *Motor Control Techniques for Figure Animation.* IEEE Computer Graphics and Applications, November 1982, 2(9), pp. 53-59.

[21] Zeltzer, D. and M. Johnson, *Motor Planning: Specifying and Controlling the Behavior of Autonomous Animated Agents.* Journal of Visualization and Computer Animation, April-June 1991, 2(2), pp. 74-80.